

AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph on page 13, line 15 to page 17, line 2 as follows:

Hereinafter, the procedure of main data error correction will be described in detail using a flowchart shown in figure 5. Initially, erasure position information about all byte positions in a code line 0 is set, and the number of erased data S in the code line 0 is counted (step S104). Before performing step S104, an error correction incapability flag indicating whether the code line is incapable of being subjected to error correction or not is initialized (step S103). When the number of erased data S counted in step S104 is equal to or smaller than 32, error correction is carried out using the erasure position information (step S106). On the other hand, when the number of erased data S is equal to or larger than 33, the error correction incapability flag is incremented from 0 to 1 (step S107), and error correction is carried out without using the erasure position information (step S108). The reason is as follows. As shown in figure 4, since, in the ECC block, the parity data area has 32 bytes, error correction can be carried out using the erasure position information when the number of erased data S is equal to or smaller than 32, but error correction cannot be carried out using the erasure position information when the number of erased data S is equal to or larger than 33. Next, the number of code lines on which error correction has been completed is incremented by 2 (step S109). That is, since interleaving has been done so that every other code line should be subjected to error correction, the code lines are rearranged in the error correction order. That is, after performing error correction on the code line 0, even-numbered code lines (code lines 2, 4, 6, 8, ..., ~~308~~302) are subjected to error correction, and thereafter, odd-numbered code lines (code lines 1, 3, 5, 7, ..., 303) are subjected

to error correction. When the code lines are rearranged in the error correction order, the code line 1 becomes the ~~152nd~~ 153rd code line counting from the 1st code line. Since, in this embodiment, error correction is performed on the ECC block shown in figure 4, the number of code lines is incremented by 2 in step S109. However, the number of code lines to be incremented depends on how many code lines have been skipped when the code lines to be subjected to error correction are arranged in the error correction order. For example, when every third code lines are to be subjected to error correction, the number of code lines is incremented by 3 in step S109. After step S109, if the number of code lines incremented is 305 ($n = 305$), it is judged that setting of erasure position information is done for all of the code lines (step S110). On the other hand, when the number of code lines incremented is not 305, it is judged whether setting of erasure position information for the even-numbered code lines has been completed or not (step S111). When the result of the judgement in step S111 is "yes", setting of erasure position information for all byte positions in the code line 1 is started. On the other hand, when the result of the judgement in step S111 is "No", it is judged as to whether the previously error-corrected code line was incapable of being subjected to error correction or not (step S113). In this first embodiment, when the immediately previous code line (i.e., the code line 0 when the result of increment in step S109 is $n = 2$) is judged as to whether it is an error-incorrectable code line or not. When the result of the judgement in step S113 is "Yes", steps S103 to S108 are repeated to set erasure position information for the target code line and, further, the number of erased data is counted. On the other hand, when the result of the judgement in step S113 is "No", all of the byte positions in the target code line are subjected to judgement as to whether the byte

positions are boundaries with the sub data area or the SY area (step S115), starting from the byte position $i = 0$ (step S114). The reason is as follows. Since the main data between sub data or the main data between sub data and SY have the same erasure position information, erasure position information should be set at only the boundary between the main data area and the sub data area or the SY area. To be specific, when the code lines are rearranged in the error correction order, the byte positions of the code line 0, code line 38, code line 76, code line 114, code line 152, code line 190, code line 228, and code line 266 are boundaries with the sub data area or the SY area. When the result of the judgement in step S115 is "No", since the erasure position information at the same byte position in the previous code line is used, the processing goes to step S119 to judge whether the next byte position is a boundary with the sub data area or the SY area. On the other hand, when the result of the judgement in step S115 is "Yes", i.e., when the byte position is a boundary with the sub data area, it is judged whether the erasure position information in the target byte position in the target code line indicates "erasure" or not (step S116). When the result of the judgement in step S116 indicates "erasure", the number of erased data is incremented (step S117). When it does not indicate "erasure", the number of erased data is decremented (step S118). The operations of the above-mentioned steps S115 to S118 are repeated up to the final byte in one code line ($i = 248$) (step S119). When erasure information setting is completed up to the final byte in the one code line (step S120), the processing goes to step S105 to execute error correction.